**Информационные системы управления корпоративным контентом**

**Практическая работа №12**

**ФИО:** Гусев Тимофей Андреевич

**Группа:** ИКБО-08-21

**Дата предоставления в СДО:**

**ЗАДАНИЕ 1. Доработать страницу отдельной накладной: ➔ реализовать необходимые таблицы для хранения товарных позиций по каждой накладной;**

**➔ отображать список товарных позиций на странице каждой накладной;**

**➔ реализовать функционал кнопки «Выдать товары по накладной» с изменением статуса и времени выдачи в соответствующей таблице базы данных.**

На листинге 1 представлен код HTML-страницы, хранящей таблицу товарных позиций.

Листинг 1 – Код файла item.html

{% extends 'base.html' %}  
  
{% block content %}  
 <h2>{% block title %} {{ item['title'] }} {% endblock %}</h2>  
 <span class="badge badge-primary">{{ item['company'] }}</span>  
 {% if item['status']=='выдан' %}  
 <span class="badge badge-success"> выдан </span>  
 {% else %}  
 <span class="badge badge-danger"> не выдан </span>  
 {%- endif %}

<hr>  
 <table class="table">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Позиция</th>  
 <th scope="col">Единица измерения</th>  
 <th scope="col">Количество</th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for pos in item["position"] %}  
 <tr>  
 <td> {{ pos[0] }} </td>  
 <td> {{ pos[1] }} </td>  
 <td> {{ pos[2] }} </td>  
 <td> {{ pos[3] }} </td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
 {% if item['status']=='выдан' %}  
 <p><h5>Товары были выданы {{ item['date'] }}</h5></p>  
 {% else %}  
 <p>  
 <form method="POST">  
 <button type="submit" class="btn btn-primary  
 btn-sm" value="give\_prod" name="give\_prod">Выдать товары по накладной</button>  
 </form>  
 </p>  
 {%- endif %}  
 </p>  
  
{% endblock %}

На листинге 2 представлен код функции, отвечающей за работу страниц с товарными позициями.

Листинг 2 – Код функции item

@app.route('/<int:item\_id>', methods=['GET', 'POST'])  
def item(item\_id):  
 if request.method == 'POST':  
 if request.form.get('give\_prod') == 'give\_prod':  
 change\_status(item\_id)  
 elif request.method == 'GET':  
 pass  
 item = get\_item(item\_id)  
 return render\_template('item.html', item=item)

На листинге 3 представлен код получения информации по накладной из базы данных.

Листинг 3 – Код функции get\_item

def get\_item(item\_id):  
 conn = get\_db\_connection()  
 item = conn.execute('SELECT \* FROM items WHERE id = ?', (item\_id,)).fetchone()  
 tmp = {}  
 for key in item.keys():  
 if key == "position":  
 tmp[key] = json.loads(item[key])  
 else:  
 tmp[key] = item[key]  
 conn.close()  
 if item is None:  
 abort(404)  
 return tmp

На рисунке 1 представлен внешний вид страницы накладной.

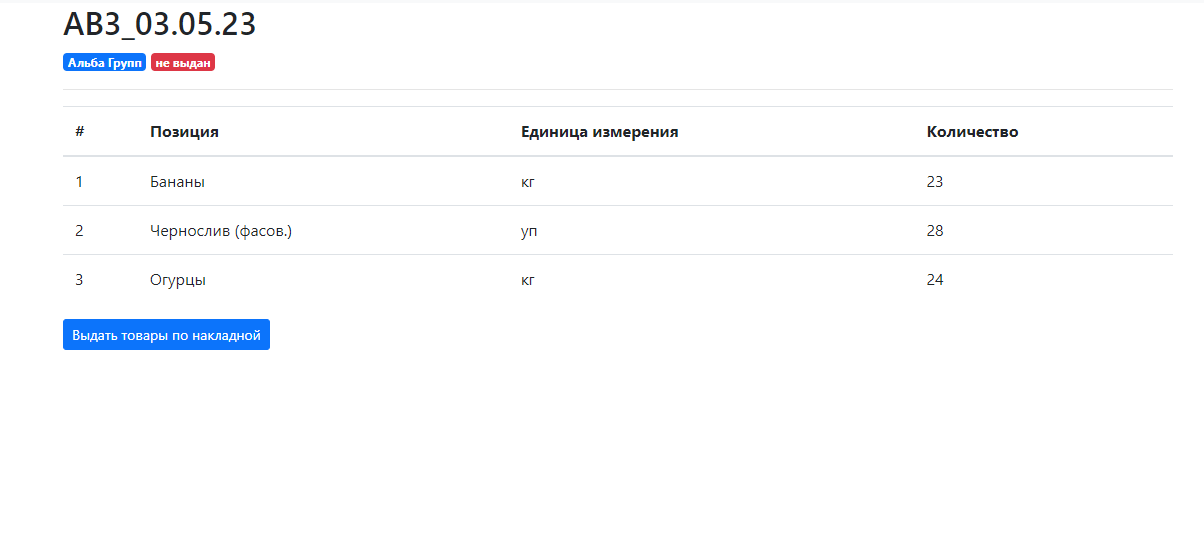


Рисунок 1 – Внешний вид страницы накладной

**ЗАДАНИЕ 2. Реализовать страницу добавления новой накладной:**

**➔ реализовать необходимые таблицы для хранения существующих товарных позиций и существующих контрагентов, с которыми ведется продажа;**

**➔ реализовать возможность выбора грузополучателя, даты выдачи и необходимого числа товарных позиций.**

На листинге 4 представлен код HTML-страницы создания новой накладной.

Листинг 4 – Код файла new\_item.html

{% extends 'base.html' %}  
  
{% block content %}  
 <h2>Создать новую накладную</h2>  
 <form method="POST">  
 <p>Грузополучатель</p>  
 <p>  
 <input type="search" list="character" name="companies" class="form-control">  
 <datalist id="character">  
 {% for com in comps %}  
 <option value="{{ com }}"></option>  
 {% endfor %}  
 </datalist>  
 </p>  
 <p>Дата</p>  
 <p>  
 <input type="date" id="date" name="date" class="form-control"/>  
 </p>  
 <p><b>Товары</b></p>  
 <table class="table">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Позиция</th>  
 <th scope="col">Количество</th>  
 </tr>  
 </thead>

<tbody>  
 {% for i in range(1, 6) %}  
 <tr>  
 <td> {{ i }} </td>  
 <td>  
 <input type="search" list="poses" name="pos{{ i }}" class="form-control">  
 <datalist id="poses">  
 {% for p in positions %}  
 <option value="{{ p }}"></option>  
 {% endfor %}  
 </datalist>  
 </td>  
 <td>  
 <input type="number" step="1" min="0" name="count{{ i }}" class="form-control"/>  
 </td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
 <p>  
 <button type="submit" class="btn btn-primary  
 btn-sm" value="create" name="create">  
 Создать  
 </button>  
 </p>  
 </form>  
{% endblock %}

На листинге 5 представлен код функции, отвечающей за работу страницы создания новой накладной.

@app.route('/new', methods=['GET', 'POST'])  
def new\_item():  
 if request.method == 'POST':  
 if request.form.get("companies") == "":  
 company = None  
 else:  
 company = request.form.get("companies")  
 if request.form.get("date") == "":  
 date = None  
 else:  
 date = datetime.strptime(str(request.form.get("date")), '%Y-%m-%d').strftime("%d.%m.%Y")  
 comps = get\_all\_companies()  
 positions = get\_all\_positions()  
 return render\_template('new\_item.html', comps=comps, positions=positions)

На листинге 6 представлены две функции по получению данных из базы данных.

Листинг 6 – Код функций get\_all\_companies, get\_all\_positions.

def get\_all\_positions():  
 df = pd.read\_excel("data.xlsx")  
 return list(set(list(df["Товар"])))  
  
  
def get\_all\_companies():  
 conn = get\_db\_connection()  
 elements = conn.execute('SELECT company FROM items').fetchall()  
 res = list(set([el["company"] for el in elements]))  
 return res

На рисунке 2 представлен внешний вид страницы после всех доработок.

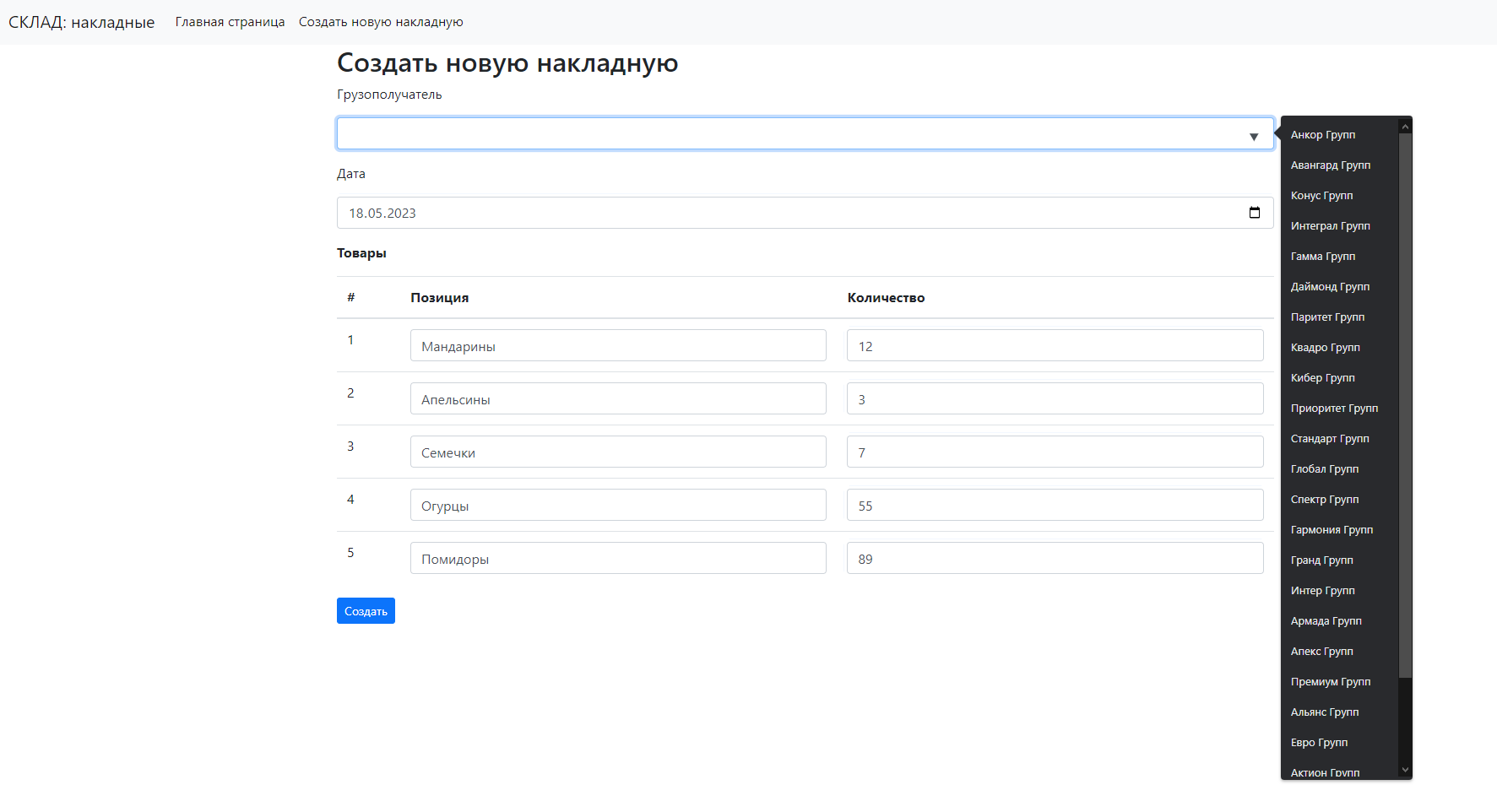


Рисунок 2 – Страница создания новой накладной